

¿Miguel entendió la clase?

Miguel estudió esta clase cuidadosamente y tomó las notas que se encuentran a continuación. ¿Puedes tu, un erudito, indicar qué declaraciones escribió Miguel incorrectamente a través de las alternativas a continuación?

- 1) Existe una gran jerarquía de clases que representan excepciones. Por ejemplo, *ArithmeticException* es hija de *RuntimeException*, que hereda de *Exception*, que a su vez es hija de la clase de excepciones más ancestral, *Throwable*. Conocer bien esta jerarquía significa saber cómo usar las excepciones en su aplicación.
- 2) *Throwable* es la clase que necesita ser extendida para poder lanzar un objeto en la pila (usando la palabra reservada *throw*)
- 3) La jerarquía que comenzó con la clase *Throwable* se divide en excepciones y errores. Las excepciones se utilizan en códigos de aplicación. Los errores son utilizados exclusivamente por la máquina virtual. Las clases que heredan de *ExceptionError* se utilizan para informar errores en la máquina virtual. Los desarrolladores de aplicaciones no deben crear errores que hereden de *Error*
- 4) *StackOverflowError* es un error de la máquina virtual para informar que la pila de ejecución no tiene más memoria.
- 5) Las excepciones se dividen en dos grandes categorías: las que el compilador comprueba obligatoriamente y las que no. Los primeros se denominan *checked* y se crean al pertenecer a una jerarquía que no pasa por *RuntimeException*. Los segundos son *unchecked* y se crean como descendientes de *RuntimeException*.

A

Miguel se equivocó en los enunciados 3 y 5



Miguel se equivocó en el enunciado 3.



Correcto. ¡Eso ahí! La primera oración del enunciado 3 es correcta. Sin embargo, las clases que informan de errores de máquinas virtuales heredan de Error. ¡Dile a Miguel que vuelva a ver los videos!

C

Ninguna. Todas las declaraciones de Miguel son correctas.



PRÓXIMA ACTIVIDAD